K 6

```
FFFFFFFFFFFFFFF     000000000      RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFFF     000000000      RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFFF     000000000      RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFF              000         000   RRR         RRR  RRR         RRR        TTT        LLL
FFF              000         000   RRR         RRR  RRR         RRR        TTT        LLL
FFF              000         000   RRR         RRR  RRR         RRR        TTT        LLL
FFF              000         000   RRR         RRR  RRR         RRR        TTT        LLL
FFF              000         000   RRR         RRR  RRR         RRR        TTT        LLL
FFFFFFFFFFFF     000         000   RRRRRRRRRRRR     RRRRRRRRRRRR           TTT        LLL
FFFFFFFFFFFF     000         000   RRRRRRRRRRRR     RRRRRRRRRRRR           TTT        LLL
FFFFFFFFFFFF     000         000   RRRRRRRRRRRR     RRRRRRRRRRRR           TTT        LLL
FFF              000         000   RRR   RRR        RRR   RRR              TTT        LLL
FFF              000         000   RRR   RRR        RRR   RRR              TTT        LLL
FFF              000         000   RRR   RRR        RRR   RRR              TTT        LLL
FFF              000         000   RRR      RRR     RRR      RRR           TTT        LLL
FFF              000         000   RRR      RRR     RRR      RRR           TTT        LLL
FFF              000         000   RRR      RRR     RRR      RRR           TTT        LLL
FFF                 000000000      RRR         RRR  RRR         RRR        TTT        LLLLLLLLLLLLLLL
FFF                 000000000      RRR         RRR  RRR         RRR        TTT        LLLLLLLLLLLLLLL
FFF                 000000000      RRR         RRR  RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
                                                                   G  5
**FILE**ID**FORVM
```

```
FFFFFFFFFF    000000    RRRRRRRR    VV      VV  MM      MM
FFFFFFFFFF    000000    RRRRRRRR    VV      VV  MM      MM
FF          00      00  RR      RR  VV      VV  MMMM  MMMM
FF          00      00  RR      RR  VV      VV  MMMM  MMMM
FF          00      00  RR      RR  VV      VV  MM  MM  MM
FF          00      00  RR      RR  VV      VV  MM  MM  MM
FFFFFFFF    00      00  RRRRRRRR    VV      VV  MM      MM
FFFFFFFF    00      00  RRRRRRRR    VV      VV  MM      MM
FF          00      00  RR  RR      VV      VV  MM      MM
FF          00      00  RR  RR      VV      VV  MM      MM
FF          00      00  RR    RR      VV  VV    MM      MM      ....
FF          00      00  RR    RR      VV  VV    MM      MM      ....
FF            000000    RR      RR      VV      MM      MM      ....
FF            000000    RR      RR      VV      MM      MM      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE FOR$$VM (              ! Internal FORTRAM Virtual memory allocation/deallocation
0         IDENT = '1-001'  ! File: FORVM.B32
0                 ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1 !++
1 ! FACILITY:FORTRAN support library
1 !
1 ! ABSTRACT:
1
1 !       Dynamic virtual memory allocation and deallocation.
1 !       FORTRAN interface with LIB$GET_VM and LIB$FREE_VM
1 !       resourec allocation procedures.
1 !
1 ! ENVIRONMENT: User access mode; mixture of AST level or not.
1 !              This module is both shared and non-shared.  Hence all
1 !              EXTERNAL references are of type GENERAL to prevent data truncation errors
1 !              when linking with the non-shared FORTRAN compatibility routines.
1 !
1 ! AUTHOR:       T. Hastings, CREATION DATE: 4-Dec-77; Version 01
1 !
1 ! MODIFIED BY:
1 !
1 ! 01     - original
1 ! 0-2    - Use FOR$$SIG_FATINT.  TNH 5-Dec-77
1 ! 0-3    - Don't clear memory.  TNH 8-Dec-77
1 ! 0-04   - Change REQUIRE files for VAX system build.  DGP 28-Apr-78
1 ! 0-05   - Add optional second arg (FCB only).  TNH 22-MAY-78
1 ! 0-06   - Use FOR$$SIG_DATCOR instead of FOR$$SIG_FATINT.  TNH 10-June-78
1 ! 0-07   - Make all external references GENERAL, since this module
1 !        - is both shared and non-shared.  TNH 3-Aug-78
1 ! 0-08   - Change file name to FORVM.B32, and change the names of
1 !            the REQUIRE files similarly.  JBS 14-NOV-78
```

```
:   58          0058  1 ! 1-001 - Update version number and copyright notice.  JBS 16-NOV-78
:   59          0059  1 !--
```

```
;   61          0060  1 !
;   62          0061  1 ! TABLE OF CONTENTS:
;   63          0062  1 !
;   64          0063  1
;   65          0064  1 FORWARD ROUTINE
;   66          0065  1         FOR$$GET_VM,            ! Allocate virtual memory - interface
;   67          0066  1         FOR$$FREE_VM: NOVALUE;  ! Deallocate virtual memory - interface
;   68          0067  1
;   69          0068  1
;   70          0069  1 !
;   71          0070  1 ! INCLUDE FILES:
;   72          0071  1 !
;   73          0072  1
;   74          0073  1         REQUIRE 'RTLML:FORERR';         ! FORTRAN error numbers
;   75          0141  1         REQUIRE 'RTLIN:RTLPSECT';       ! Define DECLARE_PSECTS macro
;   76          0236  1
;   77          0237  1 !
;   78          0238  1 ! MACROS:
;   79          0239  1 !
;   80          0240  1 !       NONE
;   81          0241  1 !
;   82          0242  1 ! EQUATED SYSMBOLS:
;   83          0243  1 !
;   84          0244  1
;   85          0245  1
;   86          0246  1 !
;   87          0247  1 ! PSECT DECLARATIONS:
;   88          0248  1 !
;   89          0249  1
;   90          0250  1         DECLARE_PSECTS (FOR);   ! declare PSECTs for FOR$ facility
;   91          0251  1
;   92          0252  1 !
;   93          0253  1 ! OWN STORAGE:
;   94          0254  1 !
;   95          0255  1
;   96          0256  1 ! EXTERNAL REFERENCES:
;   97          0257  1 !
;   98          0258  1 EXTERNAL ROUTINE
;   99          0259  1
;  100          0260  1 !+
;  101          0261  1 ! MAINTENANCE NOTE:  Since this module is called by FORTRAN compatibility
;  102          0262  1 ! routines which are un-shared and the entry points are not vectored,
;  103          0263  1 ! a separate copy of this module is linked with the user program when
;  104          0264  1 ! the user calls a FORTRAN compatibility routine.  In order to prevent
;  105          0265  1 ! data truncation errors from the linker, all external references are
;  106          0266  1 ! of addressing mode general (rather than word displacement) even for
;  107          0267  1 ! the same PSECT.
;  108          0268  1 !-
;  109          0269  1
;  110          0270  1         FOR$$SIGNAL_STO: ADDRESSING_MODE (GENERAL) NOVALUE,     ! FORTRAN SIGNAL_STOP for current unit
;  111          0271  1         FOR$$SIG_DATCOR: ADDRESSING_MODE (GENERAL) NOVALUE,     ! FORTRAN SIGNAL_STOP OTS$_INTDATCOR
;  112          0272  1                                                                 ! INTERNAL DATA CORRUPTED IN RUN-TIME LIBRAR
;  113          0273  1         FOR$$SIG_NO_LUB: ADDRESSING_MODE (GENERAL) NOVALUE,     ! FORTRAN SIGNAL_STOP when no current LUB/IS
;  114          0274  1         LIB$GET_VM: ADDRESSING_MODE (GENERAL),  ! LIBRARY allocate virtual memory
;  115          0275  1         LIB$FREE_VM: ADDRESSING_MODE (GENERAL); ! LIBRARY deallocate virtual memory
;  116          0276  1
;   118          0277   1 GLOBAL ROUTINE FOR$$GET_VM (        ! Allocate dynamic virtual memory
;   119          0278   1         NUM_BYTES,                  ! longword size in bytes
;   120          0279   1         LOGICAL_UNIT)               ! optional logical unit (if LUB/ISB/RAB not allocated)
;   121          0280   1         =
;   122          0281   1
;   123          0282   1 !++
;   124          0283   1 ! FUNCTIONAL DESCRIPTION:
;   125          0284   1 !
;   126          0285   1 !       Allocates n virtually contiguous bytes at an arbitrary  place  in
;   127          0286   1 !       the  program  region and returns the virtual address of the first byte.
;   128          0287   1 !       See description of library LIB$GET_VM for details.
;   129          0288   1 !       This procedure is provided only for convenience to FORTRAN support library.
;   130          0289   1 !       It checks for errors and SIGNAL_STOPs any.
;   131          0290   1 !       It does not clear core for speed.
;   132          0291   1 !
;   133          0292   1 !  CALLING SEQUENCE:
;   134          0293   1 !
;   135          0294   1 !       ALLOC_ADR.wa.v = FOR$$GET_VM (NUM_BYTES.rlu.v [, logical_unit.rlu.v])
;   136          0295   1 !
;   137          0296   1 !  INPUT PARAMETERS:
;   138          0297   1 !
;   139          0298   1 !       num_bytes is  an  unsigned   longword   integer value
;   140          0299   1 !                 specifying  the number of virtually contiguous bytes to
;   141          0300   1 !                 be  allocated.   Sufficient  pages  are  allocated   to
;   142          0301   1 !                 satisfy  the  request.  however, the program should not
;   143          0302   1 !                 reference  before  the  first  byte  address  assigned
;   144          0303   1 !                 (base_address)    beyond   the   last   byte   assigned
;   145          0304   1 !                 (base_adr+num_bytes - 1) since it may  be  assigned  to
;   146          0305   1 !                 another procedure.
;   147          0306   1 !
;   148          0307   1 !       [logical_unit.rlu.v] Optional logical unit number.  Used only if
;   149          0308   1 !                     an error occurs and LUB/ISB/RAB is not already allocated.
;   150          0309   1 !
;   151          0310   1 !  OUTPUT PARAMETERS:
;   152          0311   1 !
;   153          0312   1 !       None.
;   154          0313   1 !
;   155          0314   1 !  IMPLICIT INPUTS:
;   156          0315   1 !
;   157          0316   1 !       OTS$$A_CUR_LUB contains the address of the current LUB/ISB/RAB
;   158          0317   1 !       for which any errors detected will be signaled.
;   159          0318   1 !       See also LIB$GET_VM.
;   160          0319   1 !
;   161          0320   1 !  IMPLICIT OUTPUTS:
;   162          0321   1 !
;   163          0322   1 !       See LIB$GET_VM.
;   164          0323   1 !
;   165          0324   1 !  FUNCTION VALUE:
;   166          0325   1 !
;   167          0326   1 !       The address of the block allocated is returned
;   168          0327   1 !       as the function value.
;   169          0328   1 !
;   170          0329   1 !  SIDE EFFECTS:
;   171          0330   1 !
;   172          0331   1 !       The following errors are SIGNAL_STOPped:
;   173          0332   1 !
;   174          0333   1 !       FOR$_INSVIRMEM indicates 'INSUFFICIENT VIRTUAL MEMORY' when the
;  175          0334  1 !          program
;  176          0335  1 !          region was attempted to be expanded.
;  177          0336  1 !          OTS$_INTDATCOR indicates 'BAD BLOCK SIZE either 0 oor
;  178          0337  1 !          larger than FOR$K_MXVMBLK.
;  179          0338  1 !          No partial assignment is made.
;  180          0339  1 !          An appropriate number of virtual byes are removed from the  image
;  181          0340  1 !          free  memory  list.   If needed the program region is expanded by
;  182          0341  1 !          calling the SYS$EXPREG system service.  if too large  a  size  is
;  183          0342  1 !          requested or the program region could not be expended as needed.
;  184          0343  1 !--
;  185          0344  1
;  186          0345  2      BEGIN
;  187          0346  2      BUILTIN
;  188          0347  2          ACTUALCOUNT;
;  189          0348  2      LOCAL
;  190          0349  2          TEMP_ADR;         ! Adr. of block allocated
;  191          0350  2      IF NOT LIB$GET_VM (NUM_BYTES, TEMP_ADR)
;  192          0351  2      THEN
;  193          0352  3          BEGIN
;  194          0353  3          IF ACTUALCOUNT() GTRU 1
;  195          0354  3          THEN
;  196          0355  3              FOR$$SIG_NO_LUB (FOR$K_INSVIRMEM, .LOGICAL_UNIT)
;  197          0356  3          ELSE
;  198          0357  3              FOR$$SIGNAL_STO (FOR$K_INSVIRMEM)
;  199          0358  2          END;
;  200          0359  2      RETURN .TEMP_ADR;
;  201          0360  1      END;                    !  end of FOR$GET_VM routine


                                                              .TITLE  FOR$$VM
                                                              .IDENT  \1-001\

                                                              .EXTRN  FOR$$SIGNAL_STO
                                                              .EXTRN  FOR$$SIG_DATCOR
                                                              .EXTRN  FOR$$SIG_NO_LUB
                                                              .EXTRN  LIB$GET_VM, LIB$FREE_VM

                                                              .PSECT  _FOR$CODE,NOWRT,  SHR,  PIC,2

                                         0000 00000           .ENTRY  FOR$$GET_VM, Save nothing            ; 0277
                5E                   04    C2 00002           SUBL2   #4, SP                               ;
                                     5E    DD 00005           PUSHL   SP                                   ; 0350
                             04      AC    9F 00007           PUSHAB  NUM_BYTES                            ;
      00000000G   00                 02    FB 0000A           CALLS   #2, LIB$GET_VM                       ;
                  1C                 50    E8 00011           BLBS    R0, 2$                               ;
                  01                 6C    91 00014           CMPB    (AP), #1                             ; 0353
                                     0E    1B 00017           BLEQU   1$                                   ;
                             08      AC    DD 00019           PUSHL   LOGICAL_UNIT                         ; 0355
                                     29    DD 0001C           PUSHL   #41                                  ;
      00000000G   00                 02    FB 0001E           CALLS   #2, FOR$$SIG_NO_LUB                  ;
                                     09    11 00025           BRB     2$                                   ;
                                     29    DD 00027  1$:      PUSHL   #41                                  ; 0357
      00000000G   00                 01    FB 00029           CALLS   #1, FOR$$SIGNAL_STO                  ;
                  50                 6E    D0 00030  2$:      MOVL    TEMP_ADR, R0                         ; 0359
                                           04 00033           RET                                          ; 0360

; Routine Size:  52 bytes,    Routine Base:  _FOR$CODE + 0000
```

```
1 GLOBAL ROUTINE FOR$$FREE_VM (               ! Internal FORTRAN deallocate virtual memory
1                 NUM_BYTES,                  ! size in bytes
1                 BASE_ADR)                   ! Adr. of block to be deallocated
1         : NOVALUE        =
1
1 !++
1 !  FUNCTIONAL DESCRIPTION:
1 !
1 !       Deallocates  n  virtually  contiguous  bytes  starting   at   the
1 !       specified   virtual   address.  The  number  of  bytes  actually
1 !       allocated is rouned up so that the smallest number of whole  quad
1 !       words  are  de-allocated.  Numerous error checks are made to make
1 !       sure that the block being returned is a legimate free area.
1 !
1 !  CALLING SEQUENCE:
1 !
1 !       CALL FOR$$FREE_VM(num_bytes.rlu.v, base_adr.ra.v)
1 !
1 !  INPUT PARAMETERS:
1 !
1 !       num_bytes is  an  unsigned   integer
1 !                 specifying  the number of virtually contiguous bytes to
1 !                 be deallocated.
1 !
1 !       base_adr  is the address of
1 !                 the first byte to be deallocated.
1 !
1 !  OUTPUT PARAMETERS:
1 !
1 !  None.
1 !
1 !  IMPLICIT INPUTS
1 !
1 !       OTS$$A_CUR_LUB contains the address of the current LUB/ISB/RAB
1 !       for which the storage is being returned.  Any errors
1 !       are signaled on the logical unit.
1 !
1 !  IMPLICIT OUTPUTS
1 !
1 !       The pages are deallocated by calling $DEALTVA.  Then  the  pages
1 !       are  marked   as  available  in  the  OWN  storage  maintained  by
1 !       LIB$GET_VM.
1 !
1 !  COMPLETION STATUS:
1 !
1 !       None.
1 !
1 !  SIDE EFFECTS:
1 !
1 !       Any errors are signal_stopped on the current logical unit.
1 !       OTS$_INTDATCOR  indicates BAD BLOCK ADDRESS
1 !
1 !    PUTS the indicated block back on the image free storage list.
1 !--
1
2     BEGIN
2
```

```
;   260          0418  2       !+
;   261          0419  2       ! Deallocate virtual memory,  SIGNAL_STOP OTS$_INTDATCOR if error
;   262          0420  2       !-
;   263          0421  2
;   264          0422  2       IF NOT LIB$FREE_VM (NUM_BYTES, BASE_ADR) THEN FOR$$SIG_DATCOR ();
;   265          0423  1       END;



                                                   0000 00000          .ENTRY  FOR$$FREE_VM, Save nothing                      ; 0361
                                       08    AC   9F 00002          PUSHAB  BASE_ADR                                        ; 0422
                                       04    AC   9F 00005          PUSHAB  NUM_BYTES                                       ;
                      00000000G  00               02  FB 00008          CALLS   #2, LIB$FREE_VM                                 ;
                                 07               50  E8 0000F          BLBS    R0, 1$                                          ;
                      00000000G  00               00  FB 00012          CALLS   #0, FOR$$SIG_DATCOR                             ;
                                                  04 00019 1$:      RET                                                     ; 0423

; Routine Size:  26 bytes,    Routine Base:  _FOR$CODE + 0034


;   266          0424  1 END
;   267          0425  0 ELUDOM




;                                    PSECT SUMMARY
;
;       Name                       Bytes                        Attributes
;
;   _FOR$CODE                            78  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)




;                                      COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:FORVM/OBJ=OBJ$:FORVM MSRC$:FORVM/UPDATE=(ENH$:FORVM)

; Size:           78 code + 0 data bytes
; Run Time:          00:03.7
; Elapsed Time:      00:12.9
; Lines/CPU Min:       6967
; Lexemes/CPU-Min: 17508
; Memory Used:  32 pages
; Compilation Complete
```

0185 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

HELP

FORWRITDF
LIS

VMSHELP
MAP

HLD
MDL

FORWRITDU
LIS

FORVECTOR
LIS

FORWRITIL
LIS

FORWRITSF
LIS

HLD

HLDMACROS
MAR

FORVM
LIS

FORWRITSN
LIS

HLD
MAP

FORWRITSU
LIS

RSXDEFS
MAR

FORUNLOCK
LIS

FORWRITDO
LIS

HELP
LIS

FORWRITIF
LIS

FORWRITIO
LIS

HLDCSTA
LIS

FORUDFWU
LIS

FORUNDERF
LIS

FORWRITSL
LIS

FORWRITSO
LIS